NOTE THAT THE FOLLOWING WAS TAKEN FROM THE DRAFT OF THE UPDATED 1090ES SARPS

2.6.2 CPR ALGORITHM PARAMETERS AND INTERNAL FUNCTIONS

The CPR algorithm shall utilize the following parameters whose values are set as follows for the Mode S extended squitter application:

a) The number of bits used to encode a position coordinate, *Nb*, is set as follows:

Nb = 17 for airborne encoding used in the ADS-B airborne position message and the TIS-B fine airborne message

Nb = 19 for surface encoding used in the ADS-B surface position message and the TIS-B fine surface position message

Nb = 14 for intent encoding

Nb = 12 for TIS-B encoding, used in the TIS-B coarse airborne position message.

Note 1.— The Nb *parameter determines the encoded position precision (approximately 5 m for the airborne encoding, 1.25 m for the surface encoding, and 41 m for the intent encoding and 164 m for the TIS-B encoding).*

b) The number of geographic latitude zones between the equator and a pole, denoted *NZ*, is set to 15.

Note 2.— The NZ *parameter determines the unambiguous airborne range for decoding (360 NM). The surface latitude/longitude encoding omits the high-order 2 bits of the 19bit CPR encoding, so the effective unambiguous range for surface position reports is 90 NM.*

The CPR algorithm shall define internal functions to be used in the encoding and decoding processes.

c) The notation **floor**(x) denotes the floor of *x*, which is defined as the greatest integer value *k* such that $k \leq x$.

*Note 3.— For example, **floor**(3.8) = 3, while **floor**(–3.8) = – 4.*

d) The notation |*x*| denotes the absolute value of *x*, which is defined as the value *x* when $x \geq 0$ and the value –*x* when $x < 0$.

e) The notation **MOD**(*x*,*y*) denotes the “modulus” function, which is defined to return the value

$$\mathrm{MOD}(x, y) = x - y \cdot \mathrm{floor}\left(\frac{x}{y}\right) \text{ where } y \neq 0 .$$

Note 4.— The value y *is always positive in the following CPR algorithms. When* x *is non-negative,* ***MOD***(x,y) *is equivalent to the remainder of* x *divided by* y. *When* x *represents a negative angle, an alternative way to calculate* ***MOD***(x,y) *is to return the remainder of* (x+360°) *divided by* y.

For example, $\mathrm{MOD}(-40°, 6°) = \mathrm{MOD}(320°, 6°) = 2°$

f) The notation **NL**(*x*) denotes the “number of longitude zones” function of the latitude angle *x*. The value returned by **NL**(*x*) is constrained to the range from 1 to 59. **NL**(*x*) is defined for most latitudes by the equation,

$$\text{NL}(lat) = \text{floor}\left(2\pi \cdot \left[\arccos\left(1 - \frac{1-\cos\left(\frac{\pi}{2 \cdot NZ}\right)}{\cos^2\left(\frac{\pi}{180°} \cdot |lat|\right)}\right)\right]^{-1}\right)$$

where *lat* denotes the latitude argument in degrees. For latitudes at or near the N or S pole, or the equator, the following points shall be defined:

For *lat* = 0 (the equator), NL = 59
For *lat* = +87 degrees, NL = 2
For *lat* = -87 degrees, NL = 2
For *lat* > +87 degrees, NL = 1
For *lat* < -87 degrees, NL = 1

Note 5.— This equation for NL( ) is impractical for real time implementation. A table of transition latitudes can be pre-computed using the following equation:

$$lat = \frac{180°}{\pi} \cdot \arccos\left(\sqrt{\frac{1-\cos\left(\frac{\pi}{2 \cdot NZ}\right)}{1-\cos\left(\frac{2\pi}{NL}\right)}}\right) \text{ for } NL = 2 \text{ to } 4 \cdot NZ - 1,$$

and a table search procedure used to obtain the return value for NL( ). The table value for NL = 1 is 90 degrees.

When using the look up table established by using the equation above, the NL value is not expected to change to the next lower NL value until the boundary (latitude established by the above equation) has actually been crossed when moving from the equator towards the pole. This convention is supported by the equation given above in subparagraph (f) which shows the absolute value of "lat" being used in the denominator which allows the equation to be used for either Northern (+) or Southern (-) latitudes. The "floor" function in the subparagraph (f) equation then forces the convention that NL value does not change until the boundary is actually crossed when moving from the equator towards the pole.

As an example, the boundary change between NL=47 and NL=48 occurs at 36.85025107593526 degrees. Precisely at that latitude, the NL=48; therefore, for latitude values equal to the boundary and below, the NL remains at 48 until the next boundary is established when NL changes from NL=48 to NL=49. Above the boundary computed at 36.85025107593526, the NL value will be NL=47 and remain at NL=47 until the next boundary is established when NL changes from NL=47 to NL=46.

The objective of all this is to maintain airborne encoding zone sizes to 360 nautical mile rectangles and surface encoding zone sizes to 90 nautical mile rectangles. As one moves from the equator to the pole, circles of latitude decrease in radius and therefore circumference. The circumference still has 360 degrees so to keep the airborne 360 nautical mile rectangles, or the surface 90 nautical mile rectangles, the NL value must decrease, but only as the boundaries are crossed as discussed in the preceding paragraphs of this guidance material.

2.6.3 CPR ENCODING PROCESS

The CPR encoding process shall calculate the encoded position values XZ_i and YZ_i for either airborne, surface intent or TIS-B latitude and longitude fields from the global position *lat* (latitude in degrees), *lon* (longitude in degrees), and the CPR encoding type *i* (0 for even format and 1 for odd format), by performing the following sequence of computations. The CPR encoding for intent shall always use the even format (i = 0), whereas the airborne, surface and TIS-B encoding shall use both even (i = 0) and odd (i = 1) formats.

a) $Dlat_i$ (the latitude zone size in the N-S direction) is computed from the equation:

$$Dlat_i = \frac{360°}{4 \cdot NZ - i}$$

b) YZ_i (the <u>Y</u>-coordinate within the <u>Z</u>one) is then computed from $Dlat_i$ and lat using separate equations:

For Nb=17 encoding:
$$YZ_i = \text{floor}\left(2^{17} \cdot \frac{\text{MOD}(lat, Dlat_i)}{Dlat_i} + \frac{1}{2}\right)$$

For Nb=19 encoding:
$$YZ_i = \text{floor}\left(2^{19} \cdot \frac{\text{MOD}(lat, Dlat_i)}{Dlat_i} + \frac{1}{2}\right)$$

For Nb=14 encoding:
$$YZ_0 = \text{floor}\left(2^{14} \cdot \frac{\text{MOD}(lat, Dlat_0)}{Dlat_0} + \frac{1}{2}\right)$$

For Nb=12 encoding:
$$YZ_i = \text{floor}\left(2^{12} \cdot \frac{\text{MOD}(lat, Dlat_i)}{Dlat_i} + \frac{1}{2}\right)$$

c) $Rlat_i$ (the latitude that a receiving ADS-B system will extract from the transmitted message) is then computed from lat, YZ_i, and $Dlat_i$ using separate equations:

For Nb=17 encoding:
$$Rlat_i = Dlat_i \cdot \left(\frac{YZ_i}{2^{17}} + \text{floor}\left(\frac{lat}{Dlat_i}\right)\right)$$

For Nb=19 encoding:
$$Rlat_i = Dlat_i \cdot \left(\frac{YZ_i}{2^{19}} + \text{floor}\left(\frac{lat}{Dlat_i}\right)\right)$$

For Nb=14 encoding:
$$Rlat_0 = Dlat_0 \cdot \left(\frac{YZ_0}{2^{14}} + \text{floor}\left(\frac{lat}{Dlat_0}\right)\right)$$

For Nb=12 encoding
$$Rlat_i = Dlat_i \cdot \left(\frac{YZ_i}{2^{12}} + \text{floor}\left(\frac{lat}{Dlat_i}\right)\right)$$

d) $Dlon_i$ (the longitude zone size in the E-W direction) is then computed from $Rlat_i$ using the equation:

$$Dlon_i = \begin{cases} \dfrac{360°}{\text{NL}(Rlat_i) - i}, & \text{when NL}(Rlat_i) - i > 0 \\ 360°, & \text{when NL}(Rlat_i) - i = 0 \end{cases}$$

Note: *When performing the NL function, the encoding process must ensure that the NL value is established in accordance with Note 5 of 2.6.2.f.*

e) XZ_i (the X-coordinate within the Zone) is then computed from *lon* and $Dlon_i$ using separate equations:

For Nb=17 encoding: $$XZ_i = \text{floor}\left(2^{17} \cdot \frac{\text{MOD}(lon, Dlon_i)}{Dlon_i} + \frac{1}{2}\right)$$

For Nb=19 encoding: $$XZ_i = \text{floor}\left(2^{19} \cdot \frac{\text{MOD}(lon, Dlon_i)}{Dlon_i} + \frac{1}{2}\right)$$

For Nb=14 encoding $$XZ_0 = \text{floor}\left(2^{14} \cdot \frac{\text{MOD}(lon, Dlon_0)}{Dlon_0} + \frac{1}{2}\right)$$

For Nb=12 encoding: $$XZ_i = \text{floor}\left(2^{12} \cdot \frac{\text{MOD}(lon, Dlon_i)}{Dlon_i} + \frac{1}{2}\right)$$

f) Finally, limit the values of XZ_i and YZ_i to fit in the 17-bit or 14-bit field allotted to each coordinate:

For Nb=17 encoding: $$YZ_i = \text{MOD}\left(YZ_i, 2^{17}\right), \quad XZ_i = \text{MOD}\left(XZ_i, 2^{17}\right)$$

For Nb=19 encoding: $$YZ_i = \text{MOD}\left(YZ_i, 2^{17}\right), \quad XZ_i = \text{MOD}\left(XZ_i, 2^{17}\right)$$

For Nb=14 encoding: $$YZ_0 = \text{MOD}\left(YZ_0, 2^{14}\right), \quad XZ_0 = \text{MOD}\left(XZ_0, 2^{14}\right)$$

For Nb=12 encoding: $$YZ_i = \text{MOD}\left(YZ_i, 2^{12}\right), \quad XZ_i = \text{MOD}\left(XZ_i, 2^{12}\right)$$

2.6.4 LOCALLY UNAMBIGUOUS CPR DECODING

The CPR algorithm shall decode a geographic position (latitude, $Rlat_i$, and longitude, $Rlon_i$,) that is locally unambiguous with respect to a reference point (lat_s, lon_s) known to be within 180 NM of the true airborne position (or within 45 NM for a surface message).

Note.— This reference point may be a previously tracked position that has been confirmed by global decoding (see 2.6.7) or it may be the own-aircraft position, which would be used for decoding a new tentative position report.

The encoded position coordinates XZ_i and YZ_i and the CPR encoding type *i* (0 for the even encoding and 1 for the odd encoding) contained in a Mode S extended squitter message shall be decoded by performing the sequence of computations given in 2.6.5 for the airborne intent and TIS-B format types and in 2.6.6 for the surface format type.

2.6.5 LOCALLY UNAMBIGUOUS AIRBORNE POSITION DECODING

The following computations shall be performed to obtain the locally ambiguous decoded latitude/longitude for the airborne, intent and TIS-B message formats. For the intent format, i l will always be set to 0 (even encoding), whereas the airborne format shall use both even (i = 0) and odd (i = 1) encoding. For the airborne and fine TIS-B formats, *Nb* shall equal 17, for the intent format, Nb shall equal 14 and for the coarse TIS-B format, Nb shall equal 12.

a) $Dlat_i$ is computed from the equation:

:

$$Dlat_i = \frac{360°}{4 \cdot NZ - i}$$

b) The latitude zone index number, j, is then computed from the values of lat_s, $Dlat_i$ and YZ_i using the equation:

$$j = \text{floor}\left(\frac{lat_s}{Dlat_i}\right) + \text{floor}\left(\frac{1}{2} + \frac{\text{MOD}(lat_s, Dlat_i)}{Dlat_i} - \frac{YZ_i}{2^{Nb}}\right)$$

c) The decoded position latitude, $Rlat_i$, is then computed from the values of j, $Dlat_i$, and YZ_i using the equation:

$$Rlat_i = Dlat_i \cdot \left(j + \frac{YZ_i}{2^{Nb}}\right)$$

d) $Dlon_i$ (the longitude zone size in the E-W direction) is then computed from $Rlat_i$ using the equation:

$$Dlon_i = \begin{cases} \dfrac{360°/k}{\text{NL}(Rlat_i) - i}, & \text{when NL}(Rlat_i) - i > 0 \\ 360°, & \text{when NL}(Rlat_i) - i = 0 \end{cases}$$

Note 1: *When performing the NL function, the encoding process must ensure that the NL value is established in accordance with Note 5 of 2.6.2.f.*

Note 2: *Simulations and actual data have demonstrated that it is possible that the encoder function and the decoder functions may inadvertently select different NL values (difference of "1") when the latitude positions used for encoding and/or decoding are extremely close to the boundary where the NL value changes as discussed in Note 5 of 2.6.2.f.*

When this occurs, the typical result is an erroneous longitude decode with the decoded target position being a few degrees away from the expected decoded longitude position. When this occurs, the receiver decoder function must recognize that the error is induced due to operation of the encoder or decoder position very close to a boundary. To correct the situation, the decoder function will change the NL value to the NL value for the next closest possible zone and retry the remainder of the decode process which follows this subparagraph. If the resultant decoded longitude is not reasonably close to the expected decoded longitude position, then the decoder function must discard the data as it is NOT reasonable.

e) The longitude zone coordinate m is then computed from the values of lon_s, $Dlon_i$, and XZ_i using the equation:

$$m = \text{floor}\left(\frac{lon_s}{Dlon_i}\right) + \text{floor}\left(\frac{1}{2} + \frac{\text{MOD}(lon_s, Dlon_i)}{Dlon_i} - \frac{XZ_i}{2^{Nb}}\right)$$

f) The decoded position longitude, $Rlon_i$, is then computed from the values of m, XZ_i, and $Dlon_i$ using the equation:

$$Rlon_i = Dlon_i \cdot \left(m + \frac{XZ_i}{2^{Nb}}\right)$$

g) If the computed value of $Rlat_i$ is consistent with the predicted position and the value of $Rlon_i$ is not, then, using the next nearest NL zone value, repeat steps (d), (e) and (f) above. If the result of the calculation is still inconsistent with the predicted position, then the decoded data shall be discarded.

2.6.6 LOCALLY UNAMBIGOUS SURFACE POSITION DECODING

The following computations shall be performed to obtain the decoded latitude and longitude for the surface position format.

a) $Dlat_i$ is computed from the equation: :

$$Dlat_i = \frac{90°}{4 \cdot NZ - i}$$

b) The latitude zone index, j, is then computed from the values of lat_s, $Dlat_i$ and YZ_i using the equation: :

$$j = \text{floor}\left(\frac{lat_s}{Dlat_i}\right) + \text{floor}\left(\frac{1}{2} + \frac{\text{MOD}(lat_s, Dlat_i)}{Dlat_i} - \frac{YZ_i}{2^{17}}\right)$$

c) The decoded position latitude, $Rlat_i$, is then computed from the values of j, $Dlat_i$, and YZ_i using the equation:

$$Rlat_i = Dlat_i \cdot \left(j + \frac{YZ_i}{2^{17}}\right)$$

d) $Dlon_i$ (the longitude zone size, in the E-W direction) is then computed from $Rlat_i$ using the equation:

$$Dlon_i = \begin{cases} \dfrac{90°}{\text{NL}(Rlat_i) - i}, & \text{when NL}(Rlat_i) - i > 0 \\ 90°, & \text{when NL}(Rlat_i) - i = 0 \end{cases}$$

Note 1: *When performing the NL function, the encoding process must ensure that the NL value is established in accordance with Note 5 of 2.6.2.f.*

Note 2: *Simulations and actual data have demonstrated that it is possible that the encoder function and the decoder functions may inadvertently select different NL values (difference of "1") when the latitude positions used for encoding and/or decoding are extremely close to the boundary where the NL value changes as discussed in Note 5 of 2.6.2.f.*

When this occurs, the typical result is an erroneous longitude decode with the decoded target position being a few degrees away from the expected decoded longitude position. When this occurs, the receiver decoder function must recognize that the error is induced due to operation of the encoder or decoder position very close to a boundary. To correct the situation, the decoder function will change the NL value to the NL value for the next closest possible zone and retry the remainder of the decode process which follows this subparagraph. If the resultant decoded longitude is not reasonably close to the expected decoded longitude position, then the decoder function must discard the data as it is NOT reasonable.

e) The longitude zone coordinate m is then computed from the values of lon_s, $Dlon_i$, and XZ_i using the equation:

$$m = \text{floor}\left(\frac{lon_s}{Dlon_i}\right) + \text{floor}\left(\frac{1}{2} + \frac{\text{MOD}(lon_s, Dlon_i)}{Dlon_i} - \frac{XZ_i}{2^{17}}\right)$$

f) The decoded position longitude, $Rlon_i$, is then computed from the values of m, XZ_i, and $Dlon_i$ using the equation:

$$Rlon_i = Dlon_i \cdot \left(m + \frac{XZ_i}{2^{17}} \right)$$

g) If the computed value of $Rlat_i$ is consistent with the predicted position and the value of $Rlon_i$ is not, then, using the next nearest NL zone value, repeat steps (d), (e) and (f) above. If the result of the calculation is still inconsistent with the predicted position, then the decoded data shall be discarded.

2.6.7 GLOBALLY UNAMBIGUOUS AIRBORNE POSITION DECODING

The CPR algorithm shall utilize one airborne-encoded "**even**" format reception (denoted XZ_0, YZ_0), together with one airborne-encoded "**odd**" format reception (denoted XZ_1, YZ_1), to regenerate the global geographic position latitude, *Rlat*, and longitude, *Rlon*. The time between the "**even**" and "**odd**" format encoded position reports shall be no longer than 10 seconds.

Note 1.— This algorithm might be used to obtain globally unambiguous position reports for aircraft out of the range of ground sensors, whose position reports are coming via satellite data links. It might also be applied to ensure that local positions are being correctly decoded over long ranges from the receiving sensor.

Note 2.— The time difference limit of 10 seconds between the even- and odd-format position reports is determined by the maximum permitted separation of 3 NM. Positions greater than 3 NM apart cannot be used to solve a unique global position. An aircraft capable of a speed of 1 850 km/h (1 000 kt) will fly about 5.1 km (2.8 NM) in 10 seconds. Therefore, the CPR algorithm will be able to unambiguously decode its position over a 10-second delay between position reports.

As airborne-format messages are initially received from a particular aircraft, if there is no established track on this aircraft, then a global decode shall be performed using even and odd format receptions, as described in this section.

Note 3.— If the aircraft has been transmitting surface format messages and their receptions were in track, then it is not necessary to use even-odd decoding. Beginning with the first individual airborne message reception, the location can be decoded using the local decode technique, based on the previous target location as the reference.

Given a 17-bit airborne position encoded in the "**even**" format (XZ_0, YZ_0) and another encoded in the "**odd**" format (XZ_1, YZ_1), separated by no more than 10 seconds (= 3 NM), the CPR algorithm shall regenerate the geographic position from the encoded position reports by performing the following sequence of steps:

a) Compute $Dlat_0$ and $Dlat_1$ from the equation: Compute $Dlat_0$ and $Dlat_1$ from the equation:

$$Dlat_i = \frac{360°}{4 \cdot NZ - i}$$

b) Compute the latitude index:

$$j = \text{floor}\left(\frac{59 \cdot YZ_0 - 60 \cdot YZ_1}{2^{17}} + \frac{1}{2} \right)$$

c) Compute the values of $Rlat_0$ and $Rlat_1$ using the following equation:

$$Rlat_i = Dlat_i \cdot \left(\text{MOD}(j, 60 - i) + \frac{YZ_i}{2^{17}} \right)$$

Southern hemisphere values of $Rlat_i$ will fall in the range from 270° to 360°. Subtract 360° from such values, thereby restoring $Rlat_i$ to the range from –90° to +90°.

d) If **NL**($Rlat_0$) is not equal to **NL**($Rlat_1$) then the two positions straddle a transition latitude, thus a solution for global longitude is not possible. Wait for positions where they are equal.

Note: *When performing the NL function, the encoding process must ensure that the NL value is established in accordance with Note 5 of 2.6.2.f. This is more important in the Global Unambiguous Decode as large longitude errors are induced if the decode function is not selecting the NL value properly as discussed in Note 5 of 2.6.2.f. Simulations and actual data have shown that the resultant receiver decoded longitude position could actually be on the other side of the world from where the longitude position is expected to be. In such cases, it is the responsibility of the receiver decode function to declare such data as not being reasonable.*

e) If **NL**($Rlat_0$) is equal to **NL**($Rlat_1$) then proceed with computation of $Dlon_{i\ i}$, according to whether the most recently received airborne position message was encoded with the even format (i = 0) or the odd format (i = 1):

$$Dlon_i = \frac{360°}{n_i}$$

where n_i = greater of [NL($Rlat_i$) – i] and 1.

f) Compute m, the longitude index:

$$m = \text{floor}\left(\frac{XZ_0 \cdot (NL - 1) - XZ_1 \cdot NL}{2^{17}} + \frac{1}{2} \right)$$

where NL = NL ($Rlat_i$).

g) Compute the global longitude, $Rlon_0$ or $Rlon_1$, according to whether the most recently received airborne position message was encoded using the even format (that is, with i = 0) or the odd format (i = 1):

$$Rlon_i = Dlon_i \cdot \left(\text{MOD}(m, n_i) + \frac{XZ_i}{2^{17}} \right)$$

where n_i = greater of [NL($Rlat_i$) – i] and 1.

h) If the computed value of $Rlat_i$ is consistent with the coverage area of the receiver and the value of $Rlon_i$ is not, then, using the next nearest NL zone value, repeat steps (d), (e), (f) and (g) above. If the result of the calculation is still inconsistent with the coverage area, then the decoded data shall be discarded and a new global decode shall be performed.

2.6.8 GLOBALLY UNAMBIGOUS SURFACE POSITION DECODING

This algorithm shall utilize one CPR surface position encoded "even" format message together with one CPR surface position encoded "odd" format message, to regenerate the geographic position of the aircraft or target.

As surface-format messages are initially received from a particular aircraft, if there is no established track on this aircraft, then a global decode shall be performed using even and odd format receptions, as described in this section.

Note 1.— If the aircraft has been transmitting airborne format messages and their receptions were in track, then it is not necessary to use even-odd decoding. Beginning with the first individual surface message reception, the location can be decoded using the local-decode technique, based on the previous target location as the reference.

Note 2.— Even if the aircraft is appearing for the first time in surface format receptions, any single message could be decoded by itself into multiple locations, one being the correct location of the transmitting aircraft, and all of the others being separated by 90 NM or more from the correct location. Therefore, if it were known that the transmitting aircraft cannot be farther away than 45 NM from a known location, then the first received message could be decoded using the locally unambiguous decoding method described in Section 2.6.6. Under some circumstances it may be possible for an aircraft to be first detected when it is transmitting surface position messages farther than 45 NM away from the receiving station. For this reason, even-odd decoding is required when messages are initially received from a particular aircraft. After this initial decode, as subsequent messages are received, they can be decoded individually (without using the even-odd technique), provided that the intervening time is not excessive. This subsequent decoding is based on the fact that the aircraft location has not changed by more than 45 NM between each new reception and the previously decoded location.

The even-odd decoding process shall begin by identifying a pair of receptions, one in the even format, the other in the odd format, and whose separation in time does not exceed 25 seconds.

Note 3.— The limit of 25 seconds is based on the possible change of location within this time interval. Detailed analysis of CPR indicates that if the change of location is 0.75 NM or less, then the decoding will yield the correct location of the aircraft. To assure that the change of location is actually no larger, and considering the maximum aircraft speed of 100 kt specified for the transmission of the surface format, the combination indicates that 25 seconds will provide the needed assurance.

Given a CPR 17-bit surface position encoded in the “even” format (XZ_0, YZ_0) and another encoded in the “odd” format (XZ_1, YZ_1), separated by no more than 25 seconds, the algorithm shall regenerate the geographic position (latitude *Rlat*, and longitude *Rlon*) of the aircraft or target by performing the following sequence of steps:

a) Compute the latitude zone sizes $Dlat_0$ and $Dlat_1$ from the equation:

$$Dlat_i = \frac{90°}{60 - i}$$

b) Compute the latitude index:

$$j = floor\left(\frac{59 \cdot YZ_o - 60YZ_1}{2^{17}} + \frac{1}{2}\right)$$

c) Latitude. The following formulas will yield two mathematical solutions for latitude (for each value of i), one in the northern hemisphere and the other in the southern hemisphere. Compute the northern hemisphere solution of $Rlat_0$ and $Rlat_1$ using the following equation:

$$Rlat_i = Dlat_i\left(MOD(j, 60 - i) + \frac{YZ_i}{2^{17}}\right)$$

The southern hemisphere value is the above value minus 90 degrees.

To determine the correct latitude of the target, it is necessary to make use of the location of the receiver. Only one of the two latitude values will be consistent with the known receiver location, and this is the correct latitude of the transmitting aircraft.

d) The first step in longitude decoding is to check that the even-odd pair of messages do not straddle a transition latitude. It is rare, but possible, that NL($Rlat_0$) is not equal to NL($Rlat_{i}$). If so, a solution for longitude cannot be calculated. In this event, abandon the decoding of this even-odd pair, and examine further receptions to identify another pair. Perform the decoding computations up to this point and check that these two NL values are equal. When that is true, proceed with the following decoding steps.

Note: *When performing the NL function, the encoding process must ensure that the NL value is established in accordance with Note 5 of 2.6.2.f. This is more important in the Global Unambiguous Decode as large longitude errors are induced if the decode function is not selecting the NL value properly as discussed in Note 5 of 2.6.2.f. Simulations and actual data have shown that the resultant receiver decoded longitude position could actually be on the other side of the world from where the longitude position is expected to be. In such cases, it is the responsibility of the receiver decode function to declare such data as not being reasonable.*

e) Compute the longitude zone size $Dlon_i$, according to whether the most recently received surface position message was encoded with the even format (i = 0) or the odd format (i = 1)

$$Dlon_i = \frac{90°}{n_i}$$, where n_i is the greater of [$NL(Rlat_i) - i$] and 1.

f) Compute m, the longitude index:

$$m = floor\left(\frac{XZ_0 \cdot (NL-1) - XZ_1 \cdot NL}{2^{17}} + \frac{1}{2}\right)$$

where NL = NL ($Rlat_i$)

g) Longitude. The following formulas will yield four mathematical solutions for longitude (for each value of i), one being the correct longitude of the aircraft, and the other three separated by at least 90 degrees. To determine the correct location of the target, it will be necessary to make use of the location of the receiver. Compute the longitude, $Rion_0$ or $Rlon_1$, according to whether the most recently received surface position message was encoded using the even format (that is, with i = 0) or the odd format (i = 1):

$$Rlon_i = Dlon_i \cdot \left(MOD(m, n_i) + \frac{XZi}{2^{17}}\right)$$

where n_i = greater of [NL($Rlat_i$) – i] and 1.

h) If the computed value of $Rlat_i$ is consistent with the coverage area of the receiver and the value of $Rlon_i$ is not, then, using the next nearest NL zone value, repeat steps (d), (e), (f) and (g) above. If the result of the calculation is still inconsistent with the coverage area, then the decoded data shall be discarded and a new global decode shall be performed.

This solution for $Rlon_i$; will be in the range 0° to 90°. The other three solutions are 90°, 180°, and 270° to the east of this first solution.

To then determine the correct longitude of the transmitting aircraft, it is necessary to make use of the known location of the receiver. Only one of the four mathematical solutions will be consistent with the known receiver location, and this is the correct longitude of the transmitting aircraft.

Note.— Near the equator the minimum distance between the multiple longitude solutions is more than 5 000 NM, so there is no question as to the correct longitude. For locations away from the equator, the distance between *solutions is less, and varies according to the cosine of latitude. For example at 87 degrees latitude, the minimum distance between solutions is 280 NM. This is sufficiently large to provide assurance that the correct aircraft location will always be obtained. Currently no airports exist within 3 degrees of either pole, so the decoding as specified here will yield the correct location of the transmitting aircraft for all existing airports.*

2.6.9 CPR DECODING OF RECEIVED POSITION MESSAGES

2.6.9.1 *OVERVIEW*

The techniques described in the preceding paragraphs (locally and globally unambiguous decoding) shall be used together to decode the latitude/longitude contained in airborne, surface, intent and TIS-B position messages. The process shall begin with globally unambiguous decoding based upon the receipt of an even and an odd encoded position squitter. Once the globally unambiguous position is determined, emitter centered decoding shall be used to support subsequent decoding based upon a single position report, either even or odd encoding.

2.6.9.2 *EMITTER CENTERED LOCAL DECODING*

In this approach, the most recent position of the emitter shall be used as the basis for the local decoding.

Note.— This produces an unambiguous decoding at each update since the transmitting aircraft cannot move more than 360 NM between position updates.